TWG-98-37 W. E. Burr
8 July 1998

Multiple Algorithms and the Bridge CA Concept

Introduction

The Bridge CA concept [TWG-98-29], [TWG-98-33] has been accepted as the overall approach to building a Federal PKI out of a number of separate agency application specific or agency wide PKIs. The Bridge CA (BCA) approach also provides a path to connect the Federal PKI into the overall national and global PKI. The BCA concept has been incorporated into the FPKI CONOPS document [TWG-98-31] as well. However, the present CONOPS draft does not deal with multiple digital single algorithms in the context of the BCA. An earlier document [TWG-98-18] described an approach to multiple algorithms in a Federal PKI, however it did not specifically consider the effect of the BCA.

Terms

In this paper we use the following terms:

Authority Revocation List (ARL): An indirect CRL that lists all the revoked CA certificates for all the CAs in the FPKI.

Bridge CA (BCA): A CA that is to be a bridge of trust that provide trust paths between the various trust domains of the Federal PKI, as well as between the Federal PKI and non-federal trust domains;

certificate: A digitally signed document that binds two or more attributes together. In this paper we are only concerned with x.509 digital signature certificates that bind a subject's digital signature public key (as opposed to his key management or encryption key) to his name.

Certificate Revocation List (CRL): A signed list of certificates that have been revoked;

Certification Authority (CA): A trusted entity that issues (i.e., signs and publishes) certificates and/or CRLs;

certification path: a sequence of certificates beginning with a self-signed signature certificate issued by a CA trusted by a relying party and ending with an end-entity's signature certificate, where the issuer of any certificate in the sequence is the subject of the preceding certificate;

consistent certificate: a certificate is considered to be consistent when the same algorithm is used for the public key certified in the certificate and to sign the certificate;

end-entity: a certificate holder that is not acting as a CA. In most cases an end user with a certificate.

hybrid certificate: A certificate where the subject's algorithm for the certified key is different than the algorithm used by the issuing CA to sign the certificate.

Principal CA (PCA): A CA within a trust domain that cross-certifies with the Federal BCA. Each trust domain has one principal CA.

relying party: An entity that validates a digital signature;

self-signed certificate: A certificate signed with the key it certifies. It is used by a CA to state (but not authenticate) its public key;

trust domain: In the Federal context a trust domain is a portion of the Federal PKI that operates under the management of a single *policy management authority*. One or more Certification Authorities exist within the trust domain.

Discussion

Several digital signature algorithms will be used in the government and elsewhere and individual trust domains may standardize on different algorithms. At the present time, two main algorithms, DSA and RSA are in common use in the Federal Government, and ECDSA will probably see growth use in the future. The design of the FPKI must be general enough to accommodate these algorithms and must also allow for the introduction of new algorithms. It must provide a way for certificate holders of different algorithms to interoperate.

In principle, any digital signature algorithm can be used to sign a certificate that certifies the key for any public key algorithm (by the terminology defined above, a certificate signed by one algorithm, for a key of a different algorithm, is called a hybrid certificate). The conclusion of [TWG-98-18] and earlier TWG studies on multiple algorithm interoperability was that the best approach is an "end-entity" approach, where:

- the end-entity normally signs with a single algorithm, minimizing the number of keys and certificates he is required to hold and manage;
- in principle, any digital signature algorithm can be used to sign a certificate that certifies any key for any other algorithm;
- if the end-entity needs to interoperate with end-entities who use other algorithms, then his client should be able to validate signatures for other algorithms (but not necessarily to sign with multiple algorithms);
- therefore, in the interest of broad interoperability, Federal users should be encouraged to employ clients that can validate all the common (or FIPS approved) digital signature algorithms;
- End-entity certificates should never be hybrid certificates, since any relying party must be able to validate 2 different algorithms to validate a signature signed under that hybrid certificate (even if the relying party uses the same CA as the signatory), and that plainly increases the likelihood of algorithm interoperability problems. Moreover hybrid certificates are likely to require parameters to be carried in the certificates, which increases their size substantially, and end-entity certificates are by far the most numerous class of certificates.

The last point, in fact, can be generalized farther. It is desirable, in the interest of local interoperability, for individual trust domains to maintain signature algorithm consistency. This will minimize the number of certificates that need to explicitly state parameters in the certificates, and minimize the chances for algorithm induced interoperability problems within a trust domain.

If the CA Alice trusts signs with Algorithm "white" and Bob's key is for algorithm "gray," then, if Alice is to validate Bob's signature, there must exist a hybrid certificate for a gray algorithm key, but signed with a white algorithm key, somewhere in the certification path between them. The principle that local trust domains should minimize the use of multiple algorithms within the domain implies that the needed hybrid certificates should be associated with the BCA that connects trust domains. That is, the hybrid certificates that are needed to provide trust paths between users of different algorithms are properly either issued by or issued to the BCA.

Possible Solutions

Figure 1 illustrates three possible approaches to multiple algorithms with the BCA, for the case of two algorithms, labeled "white" and "gray." Showing only 2 algorithms helps to keep the illustration simple, but we must bear in mind that we will probably ultimately need to accommodate more than two algorithms, and the approach we adopt must consider this.

Preferred Algorithm Approach

One obvious approach, illustrated in Figure 1 (a), is to have a single "preferred" algorithm (in the figure "white") for the bridge CA. Every client must be able to validate at least this algorithm if it is to use certification paths created by the bridge CA. Domains that use the "gray" algorithm might cross certify directly with each other, or there could be a "helper" bridge CA for the Gray algorithm, that just connected gray domains. The obvious difficulty is in picking the preferred algorithm. To paraphrase Orwell, "All algorithms are equal, but some algorithms are more equal than others." If we can agree on a single Bridge CA algorithm, and everyone will implement at least the ability to validate signatures that use it, this is an efficient choice. Those who don't implement verification of the chosen algorithm will be left out. It is not clear however, how we can select a single preferred algorithm for the Bridge CA, when we cannot manage to do so for more general use.

Multiple Algorithm Bridge CA Approach

A second approach, the "multiple algorithm Bridge CA" approach, is illustrated in Figure 1(b). Here the bridge CA is able to sign with multiple algorithms. Principle CAs need only sign only with the chosen algorithm for their domain. Each Principle CA issues a consistent certificate to the bridge CA, certifying the Bridge's key for whichever algorithm the Principle CA uses. The Bridge CA issues several certificates to each Principle CA, certifying the Principle CA's key with a certificate signed by every one of the Bridge CA algorithms. If the relying party and the certificate subject use the same algorithm than an entirely consistent certification path exists. If the relying party can verify the certificate subject's algorithm and his own algorithm, then a path with one hybrid certificate exists. It is not necessary for the PCAs to create any hybrid certificates; the needed hybrid certificates can be created entirely by the BCA. There would be one ARL signed by each algorithm.

If there are n PCA's and m algorithms, the total number of hybrid certificates are $n \times (m - 1)$, and these are in addition to the certificates that would be required for the Preferred Algorithm approach. One issue would be the naming and matching rules to enable clients to find the needed certificate or cross-certificate pair from among the several issued by the BCA to each CA.

Split Bridge CA Approach

In this approach, illustrated in Figure 1 (c), the Bridge CA is decomposed into as many Bridge CAs as there are signature algorithms. Each Bridge signs with only one of the algorithms and cross-certifies only with those PCAs that use the same algorithm, using consistent certificates. The Bridge CAs cross-certify among themselves, with hybrid certificates. The only hybrid certificates would be between the components of the Federal Bridge CA. If the composite Federal Bridge CA cross certifies with other Bridge CA's it would do as it does to PCAs that is with consistent certificates.

The separate Bridge CAs are logically, but not necessarily physically, separate entities. A single CA workstation that could sign with each algorithm could be used to implement all of the BCAs. Each of the BCA components would, however have it's own name. Each BCA component would issue an ARL signed with its own algorithm.

As with the multiple algorithm approach, users of the same algorithm would never have another algorithm to validate in the certification paths between them. The major disadvantage to this approach, as opposed to the multiple algorithm bridge, is that certification paths between users of different algorithms would have one extra certificate in the certification path. However the different names of the Split BCA components should simplify finding the needed hybrid certificates. Additionally, the number of hybrid certificates would be m × (m -1), which should be fewer than needed for the multi-algorithm BCA approach (since there should be many more PCAs than algorithms).

Conclusion

The most efficient choice is to use the preferred algorithm approach, however it may be impractical to select a preferred algorithm, and the approach does not itself provide a framework for changing the preferred algorithm, when that becomes desirable. The Multiple Algorithm Bridge CA and Split Bridge CA approaches are not profoundly different. The former results in slightly shorter hybrid certification paths and the latter results in fewer extra hybrid certificates and may possibly be easier to implement, because the CA names identify the algorithm used.

This paper has postulated that trust domains would use a single algorithm. However, it should be apparent that, the same technique use with the bridge CA could be applied to a principal CA to support multiple algorithms within trust domains.

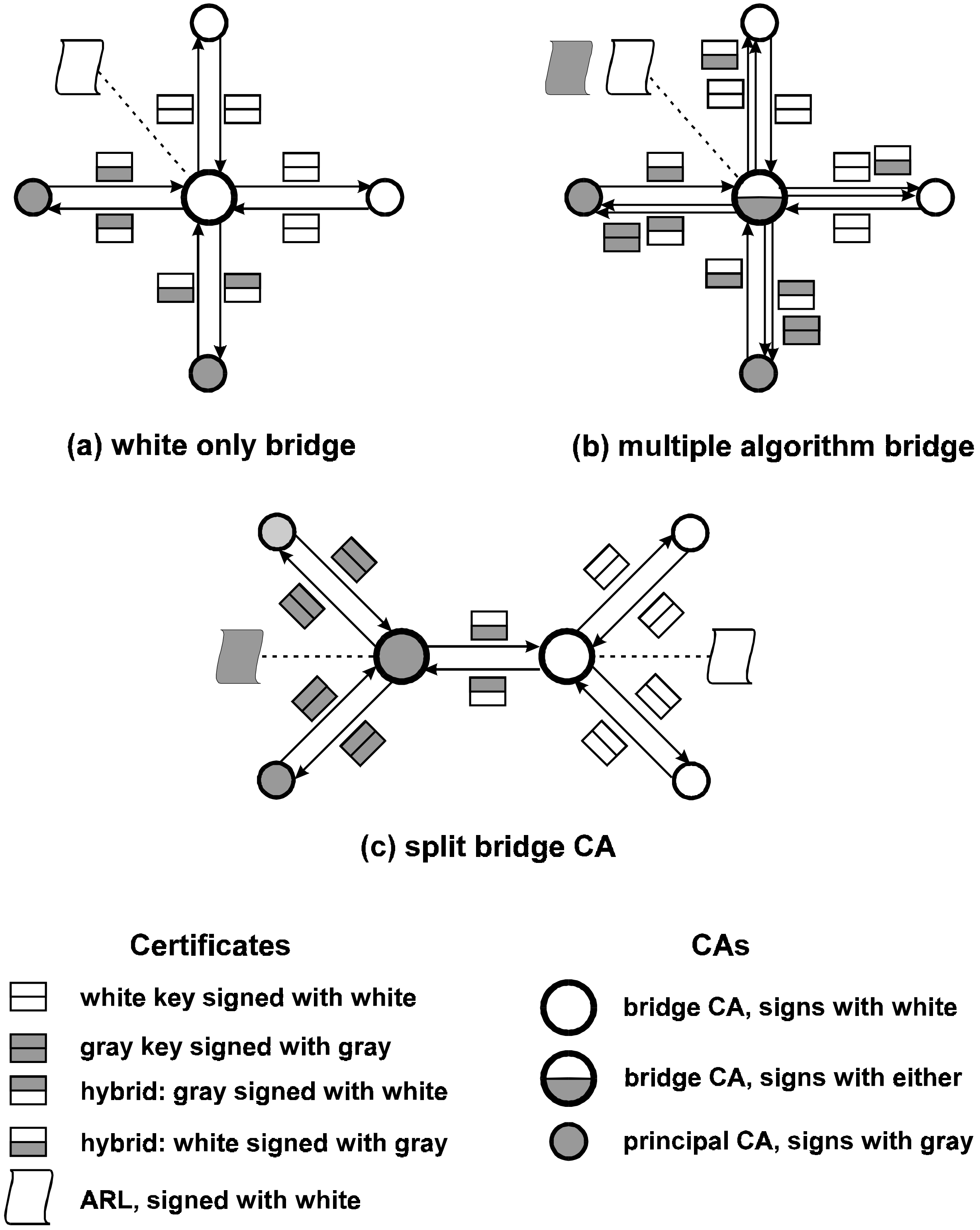
References

[TWG-98-18] W. E. Burr and W. Polk, “A Federal PKI with Multiple Digital Signature Algorithms”, April 8, 1998

[TWG-98-29] W. E. Burr, “Proposed Federal PKI Architecture,” 19 May 1998

[TWG-98-31] Draft Federal PKI Concept of Operations, 3 June 1998.

[TWG-98-33] R. Guida, “Notional Description of a Federal Policy Management Authority and Bridge Certification Authority” June 1998



(a) white only bridge

(b) multiple algorithm bridge

(c) split bridge CA

Certificates
white key signed with white
gray key signed with gray
hybrid: gray signed with white
hybrid: white signed with gray
ARL, signed with white
CAs
bridge CA, signs with white
bridge CA, signs with either
principal CA, signs with gray

Figure 1 - Multi-Algorithm BCA Architecture Choices